



# Linux VFAT kernel FS Enhancement XVFAT for kernel 2.4.20

---

2005.03.25  
machida@sm.sony.co.jp

Translation by N. Asai, IBM



# Enhancement Items

---

1. Media removal during “mount”
  - Notification of media removal to application
  - Cancellation of I/O Elevator for Block device
2. Japanese file name support
3. Dirty Flag support
4. TIME ZONE support

# Media removal during “mount”

---

- Protect data other than a file and directory entry that is actually in the middle of I/O operation by a removal of a media.
  - Manage an update of meta-data in detail
- Invalidation of cache after a removal of a media
  - Invalidation of cache of XVFAT
  - Disposal of buffer which is not written back to a physical device
- Return “error” to system calls to a file
  - Add interface to detect whether an error is caused by a media removal. -> Notification of a media removal to applications.
  - Remount from Userland.
- Currently, USB MS controller is assumed

# Media removal during “mount” - 2

---

- Block system calls until a completion of writing to a device
  - Application can know that the operation was “closed” safely
- System including XVFAT should work correctly after a removal of a media
  - (do not know right now)
  - No big problem was found so far...

# Notification of media removal to application

---

- Add ioctl for a file system
  - Can check whether a media is removed after “mount” or not
  - Can issue for a file descriptor on XVFAT

```
ioctl(fd, VFAT_IOCTL_GETMINFO, vfat_media_info*);
```

```
#define VFAT_MINFO_CFLAG_CHANGE 1
```

```
#define VFAT_MINFO_CFLAG_INSERT 2
```

```
struct vfat_media_info{
```

```
    int change;        /* 0: not ejected, 1: ejected (read only) */
```

```
    int insert;        /* 0: not inserted, 1: inserted (read only) */
```

```
    int capability;    /* capability flags (read only) */
```

```
};
```

# Notification of media removal to application - 2

---

- Notification of a media removal by proc file system
  - Status of a media
    - /proc/removabledisk/status
    - Can check status of a media any time
  - Transition of status of a media
    - /proc/removabledisk/event
    - Released (from block condition) when status is changed.

```
$ cat /proc/removabledisk/status  
sd(8,1) 1 0  
$ cat /proc/removabledisk/event  
sd(8,1) 1 1
```



# Japanese file name support

---

- One-to-One relation is not established among JIS and Unicode
  - Some characters have “N-to-1” relations
  - Conversion from JIS to Unicode, then conversion back to Unicode to JIS does not return to the same JIS code.
  - Influence on dentry cache behavior
- Ensure there is no problem even if the above problematic characters are used
  - If needed, clarify restrictions
  - Check in the case of ignoring upper case and lower case of alphabetical characters
- Provide the way to know a code conversion beforehand
- Use conversion table that is compatible to Windows.

# Japanese file name support - 2

---

- Character code conversion API for a file name of VFAT
  - Add API that converts from encode used by I/O to character code (Unicode) used by VFAT entry
  - Assume that it is used by applications that scan directory
  - ioctl to the descriptor to the “open” directory on VFAT





# Cancellation of I/O Elevator

---

- Cancel I/O Elevator to Physical Device
  - For an effectiveness of independent I/Os
  - Commonly used by block devices
  - To control a write sequence precisely from VFAT
- Function can be replaced/overwritten
  - Function can be defined by each block device
  - Currently one function is used by all devices
- Change a function from VFAT
  - Change a function by searching kernel structures of block devices.
  - Swap when “mount”, and swap back when “unmount”



# dirty flag support

---

- There is a dirty flag in FAT16,32
  - Written in an official document from Microsoft
  - Source code of “mtools” disclosed it
- Set when “mount”, and reset when “umount”
  - No more action as XVFAT
  - Assume to be used by fsck or chkdsk (Windows)

# dirty flag support - 2

---

- Special entry of FAT
  - First two entries are reserved
    - BPB\_Media bytes(0xF8)
    - Value of “End Of Cluster”
  - 2<sup>nd</sup> bit is ClnShutBitMask bit
    - 1: clean, 0: dirty
  - No unassigned bit in FAT12
- FAT format of MemoryStick
  - Different from cordial format
    - 4-64MB: FAT 12, 128MB: FAT16
  - Recorded in a media in MS Pro
    - Formatted by a media itself
    - Currently FAT16(256M-1G)



# TIME ZONE support

---

- Correct time stamp
  - FAT always records in “local time”
  - Kernel is UTC based
  - If a media from a different time zone is mounted, time stamp may not be consistent



# TIME ZONE support - 2

---

- Mount option
  - Function to add a local time at recording
- POSIX like
  - zoneinfo does not exist in kernel
    - JST,PST, etc. will be ignored.
  - If daylight saving time, need to specify it.

# TIME ZONE support - 3

---

`mount -t xvfat -o timezone=JST-9`

Time zone which is 9 hours ahead of UTC

`mount -t xvfat -o timezone=PST8PDT1,M4.1.0/2,M10.5.0/2`

Time zone which is 8 hours behind from UTC

DST starts from the first Sunday in April, and ends at the last Sunday of October, and makes one hour ahead.

DST starts at 2 am

# Others

---

- Implement as an independent File system
  - Old VFAT can be used simultaneously
  - Not a fully upward compatibility
    - Write protect of mmap(2)
    - Prohibit compressed file system plug-in (CVF)
- Dependency on device drivers of block devices
  - Need to have a capability to check a media removal
  - Currently, only USB Mass Storage has
- Removal of a device
  - If there is no device, no media existence check can be done
  - If the implementation is to remove a device if a media is removed, then, there will be no-way to work with this.